## APPENDIX D

**EXPRESS MAIL LABEL NO.: EL386270487US**

I Introduction

A camera model, as preferably implemented under the SIP in the Integrated Circuit packaging (ICP) project, is described. Two main issues are preferably resolved by the model: (a) any geometric information derived from the scanned image is preferably compensated for the distortions evolved in the image acquisition process (this is accomplished by the alignment transformation) to align the data, and (b) since several cameras acting together result in distributed and redundant data, merging and "stitching" of data is typically used.

The classes and tasks supporting the management of several cameras in the system are described, particularly new classes and tasks designated in specific to support the camera model, whereas other classes and tasks which have only a partial role in supporting the management of data in a system of several cameras, are mentioned in less detail.

2 The model

The following is a discussion of the integral components of the system:

2.1 Classes 2.1.1 Coordinate System (CoordSys)

CoordSys structure is a fundamental component of the camera model, as it preserves the integrity of coordinate data in a system of several cameras. A coordinate system representation is attached to many different items in the system; transformations (AffineSdtrana), data sources (Data-source), tasks (Sip-task), SIP data (the many Sipdata structures) and windows (Sipwin). In order to accurately reflect coordinates throughout the system processing, coordinate systems need to go through the proper transition rules whenever they are changed. Integrity is preserved by endowing any transformation method, in specific, the Transform method, which transforms an object from one coordinate system into another, with a transition rule which defines accurately the change of coordinates: Domain coordinate system -> Range coordinate system. In addition, a simple sanity check is done to make sure that the Domain coordinate system equals the coordinate system of the object to be transformed. Moreover, for a compound object, all its data members are in the same coordinate systems (e.g., all data sources of a task, or all sipdatas within a window need to be in the same coordinate system).

The CoordSys structure has three fields:

• type - can have one of the following values { Unknown, Camera, Aligned}.

• context - the context of the coordinate system represented by this object, can have one of the following values { None, Reference, Online }.

« camera-id - If Camera coordinate type, this indicates the Camera id number.

1. 1. Many items in the system are transformed by their Transform method

2.1.2 Single camera environment (Single-camera)

Single-camera class handles information on the parameters defining the environment of a single camera. To mention only the integral ones: pixel size, camera-width, the camera-coverage region (i.e., the quadrilateral aligned region "seen" by that camera), the (Affine) transformation that converts the camera coordinates into aligned coordinates, and the parameters that can be derived from it (e.g., the camera's orientation with respect to the direction of scan). This class provides several methods of inclusion tests indicating whether or not a given geometric object is embedded in the camera's coverage region, and whether or not it resides within an overlap region of the camera. To support a fine discrimination among the several different geometrical relations a geometric entity can have with respect to the camera's coverage region, Single-camera provides the WinGeomInclusion status which can be one of the following:

• ALL_tNSIDE - Entity is inside camera not intersecting any overlap region.

• ALL_IN - Entity is inside camera and intersecting an overlap region.

• CLIPPED_SHARED_LEFT - Entity is clipped against edges covered by a left overlap region.

• CLIPPED-SHAREDJR.IGHT - Entity is clipped against edges covered by a right overlap region.

• CLIPPED_N'OT_SHARED - Entity is clipped against edges not covered by an overlap region.

• ALL-OUT - Entity does not intersect the camera region.

• NOTIN-A-CAMERA - The coordinate system of the entity does not correspond to a camera.

• UNKNOWN

2.1.3 System of several cameras (Multi-cameras) The Multi-cameras class handles information on several cameras.

• Purpose: Handle information on several cameras.

• Methods:

- Put/Get number of cameras to handle.

— Put/Get a single camera information (delivery of Single-camera's methods of a specific camera, given its id number).

- Y-offset: given the id numbers of two cameras, returns the y-offset between the two cameras. — Point Inclusion (Isin): given a point data items, returns information on all cameras in the system that contains the item in their coverage regions, using Single-camera.'.'Isin for each camera. The information can be given either by a vector of cameras (or camera id numbers) containing the point, or a boolean vector indicating for each camera whether or not it contains the point.

- Polygonal Inclusion (Clip): given a quadrilateral (Dpolyline), clips it against the quadrilaterals representing the cameras coverage regions, using Single-camera::Clip for each camera.

2.1.4 Sip Camera Model (Sip-camera-model)

The Sip-camera-model class is used to handle information on a system of several cameras under the SIP. This class enables loading of camera parameters from configuration files.

2.1.5 Sip General Data (Sip-generaLdato)

Sip-general-data is a singleton aimed to provide access to general parameters and data, among which one can find access to the camera model related to the current scan, and the camera model related to the corresponding reference (methods Camera-Model() and Ref-Camera_Model(), respectively).

2.1.6 Windows (Sipwin)

A Sipwin object represents a rectangular region consisting of a variety of Sipdata items. As this object is the basic unit in the SIP, we endow windows with coordinate systems, and since a window is a geometric entity, it also endowed with WinGeomInclusion status.

2.2 Tasks 2.2.1 Reference Packing

Two kind of windows are processed during inspection: (a) top-down windows defined in aligned coordinates, and (b) on-line windows created around trigger in on-line camera coordinates. In either case, corresponding reference information need to be attached to on-line windows.

Top-Down (TD) Windows Management (Task-packer) Top-Down windows are given in aligned coordinates. For each execution graph, Task-packer creates camera coordinate TD window by first transforming the aligned window boundaries into camera coordinates. The correct transformation for the specific window is available to the Task-packer from its transformation Data-Source (ds-trans) computed for the line of window trigger. TD windows are given an id number. This id number indicates the origin of the window and will serve (at the final stage of merging information retrieved from several execution graphs) to identify windows with the same origin. TD reference information is transformed into on-line coordinates before being attached to windows and is held in that manner throughout the entire inspection processing.

Packing reference data "on the fly" (func-winref-adaptor) For on-line windows created around a trigger during inspect, relevant reference data need to be extracted from camera-based CEL reference. The on-line window boundaries determine the boundaries of the window of data to be extracted as follows: the on-line boundaries are first transformed into reference aligned coordinates. Then, for each camera in the reference camera model, those reference aligned window boundaries are transformed into camera coordinates, and the window is clipped against the camera coverage region. The camera for which the window fully falls within its coverage region, is chosen, and CEL data is extracted using the corresponding camera-based windowing query. Because CELs cannot be transformed from one coordinate system to another without a significant impairment to their structure, the extracted data is used as it is by a composite of functions comparing the on-line with the reference CEL data.

2.2.2 Single Slice Management (Task.test-manager)

Task-test-manager is traditionally responsible for the management and control of an entire execution graph (corresponding, of-course, to a single slice). In its current version, this task has an input and output queues of windows. The output queue consists of unresolved windows - which are windows that cannot be completely resolved within a single slice management and need to be directed to multi-slice processing. Task-test-manager is responsible to execute the functions attached to a window on that window, and to control their result.

2.2.3 Multi Slice Management (Task-multi-slice-manager)

Task-multLsUce-manager is responsible for the management and control of windows arriving from several slices. The main difference between Task-test-manager and Task-multi-slice-manager is that the later is responsible for collecting windows having the same id (that is, windows with the same origin) as sub-windows of an overall window, and only then to Execute the corresponding multi-slice functions attached to that window on that window, and to control their result.

2.2.4 Split of Transformation Information (Task-split-trans)

Task-split-trans splits a transformation Cameras-Online —> Aligned-Reference given for one camera into the corresponding transformations for each of the cameras available in the camera model. Online - Reference transformation should take into account two transformations: first, the alignment transformation, transforming data from camera into aligned coordinates, and the transformation resulted from the registration process, which transform online data into reference coordinates. Note: The current registration procedure finds a match between online data coming from the central camera with reference data coming from the same camera. Thus, the transformation given to the Task-split-trans as input is not Cameras-Online -> Aligned-Reference as we would expect it, but Camera(l) -Online —>• Camera(l) -Reference, and all multiplications are done accordingly.